

PROJET DE P6

Etude des anamorphoses



Etudiants :

Jeanne BERTOLDI

Robin CAMPET

Sébastien COLLET

Mickael NAVEL

Matthieu DUPREY

Paul TOURTOIS

Enseignant-responsable du projet :

Emilien VAREA

Date de remise du rapport : **18/06/12**

Référence du projet : **STPI/P6/2012 – 2013**

Intitulé du projet : **Etude des anamorphoses**

Type de projet : ***Bibliographie et Expérimentation***

Objectifs du projet (10 lignes maxi) :

Le but du projet est l'étude des anamorphoses coniques et cylindriques. Comprendre leurs fonctionnements mais également réussir à en créer sont deux des principaux objectifs de ce projet.

A partir de fonctionnement scientifique simple tel que l'optique géométrique, nous souhaitons schématiser des phénomènes présents dans de nombreux domaines scientifiques et qui paraissent compliqué.

Mots-clefs du projet (4 maxi) : ***Anamorphose, miroir, conique et cylindrique***

Table des matières

1. Introduction	5
2. Méthodologie / Organisation du travail	6
3. Travail réalisé et résultats	7
3.1. L'anamorphose conique	7
3.1.1. Un peu de culture !	7
3.1.2. Fonctionnement scientifique	8
3.1.3. Construction d'un pantographe	10
3.1.4. Programme permettant de créer ces anamorphoses	13
3.1.5. L'utilisation du logiciel Pov-Ray	14
3.2. L'anamorphose cylindrique	16
3.2.1. Fonctionnement scientifique	16
3.2.2. Programme permettant de créer ces anamorphoses	17
3.2.3. Utilisation de Pov-Ray	18
4. Conclusions et perspectives	21
4.1. Conclusions sur le travail réalisé	21
4.2. Conclusions sur l'apport personnel de cet E.C. projet	21
4.3. Perspectives pour la poursuite de ce projet	21
5. Bibliographie	22
6. Annexes	22
6.1. Enoncé des programmes utilisés	22
6.2. Listings des programmes réalisés	22
6.2.1. Calcul d'une anamorphose conique	22
6.2.2. Calcul d'une anamorphose cylindrique	25
6.2.3. Code Pov Ray en conique pour reformer l'image	26
6.2.4. Code Pov Ray en cylindrique pour reformer l'image	26

1. INTRODUCTION

Actuellement étudiant en 2^{ème} année de l'INSA Rouen, nous avons réalisé un projet de physique ayant pour but de parfaire nos connaissances dans un domaine de la physique particulier : l'optique. L'organisation du projet nous a également permis d'approfondir nos connaissances en termes de gestion de projet, travail en équipe et ainsi d'améliorer deux compétences importantes de ce type de travail : la communication et le sens de l'organisation.

Notre groupe était composé de six personnes pour étudier les anamorphoses. Malgré le peu de connaissances sur le sujet avant de démarrer le projet, notre curiosité mais aussi notre attrait pour l'optique géométrique nous ont encouragés à choisir ce sujet.

Ainsi la déformation d'image et sa reconstitution sur différents supports ou anamorphose est un sujet faisant appel à différentes notions intéressantes. Utilisée dans différents domaines de la vie quotidienne comme le cinéma, la cartographie, la télévision mais également des domaines moins communs comme pour les périscopes des chars d'assaut, l'anamorphose a une place importante dans le monde scientifique actuel.



2. METHODOLOGIE / ORGANISATION DU TRAVAIL

Durant la première séance, notre enseignant M. Varea nous a donné un aperçu de ce que nous pourrions réaliser en P-6 ce semestre. Nous nous sommes rendu compte que les anamorphoses offrent de multiples possibilités de travail. Les anamorphoses coniques, en particulier, ont été abordées plus profondément en raison de leurs relatives simplicités. C'est lors de cette séance et de celle qui suivit que nous avons trouvés nos pistes de travail : nous voulions, en plus de l'étude théorique, réaliser un objet concret. Ainsi, nous voulions réaliser un cône en texture « miroir » et un « pantographe » pour l'étude des anamorphoses coniques. Pour des raisons pratiques, l'aspect informatique du projet a été grandement développé, car nous pouvons grâce à cela obtenir rapidement un grand nombre d'images, ce qui est nécessaire pour étudier les anamorphoses.

Afin de réaliser au mieux le projet, il nous fallait nous séparer en plusieurs équipes. Le groupe étant constitué de 6 élèves, nous avons fait 3 équipes 2 élèves :

La partie mécanique : Paul et Mickael se sont occupés de réaliser un pantographe en mécano, afin que nous puissions réaliser manuellement une anamorphose d'une figure géométrique simple.

La partie informatique : Jeanne et Robin ont appris à utiliser le logiciel « Pov-Ray », qui permet la création d'images en 3D avec des formes géométriques, et l'insertion d'images numériques.

La partie programmation : Matthieu et Sébastien se sont attaqués à la programmation d'un logiciel qui, à partir d'une image numérique donnée, change celle-ci de manière à obtenir sa déformation via une anamorphose conique.

L'objectif de ces travaux est la maîtrise complète de l'anamorphose conique : pouvoir déformer une image réelle avec le pantographe et observer le résultat avec le miroir conique, et de la même manière pouvoir déformer une image numérique avec notre logiciel et observer l'image obtenue avec « Pov-Ray ».

Par la suite, nous nous sommes intéressés à un autre type d'anamorphose : l'anamorphose cylindrique. L'expérience et la méthode de travail acquises avec nos études sur l'anamorphose conique nous a permis de gagner beaucoup de temps et de comprendre le principe de cette anamorphose facilement, chaque sous-groupe s'étant spécialisé dans un domaine.



3. TRAVAIL REALISE ET RESULTATS

3.1. L'anamorphose conique

3.1.1. *Un peu de culture !*

Une anamorphose est par définition une déformation réversible d'une image à l'aide d'un système optique ou même d'un procédé mathématique (voir plus loin). Dans d'autres cas, celle-ci désigne également la déformation d'une image de télévision afin de l'adapter à un écran électronique (prenons par exemple le 16/9). Cette dernière est effectuée à l'aide d'un système optique ou électronique.

D'un point de vue artistique, de nombreux personnages célèbres ont proposé et produit des œuvres grâce à ce procédé. Ainsi, on peut remarquer plusieurs grandes œuvres d'anamorphoses comme la célèbre toile Les Ambassadeurs de Hans Hobein. Elles sont également utilisées comme trompe-l'œil architecturale ou même pour le plaisir de certains artistes peu connus.



Figure 1 : Exemple d'un trompe-œil anamorphose (La gargouille n'est en fait pas une sculpture)

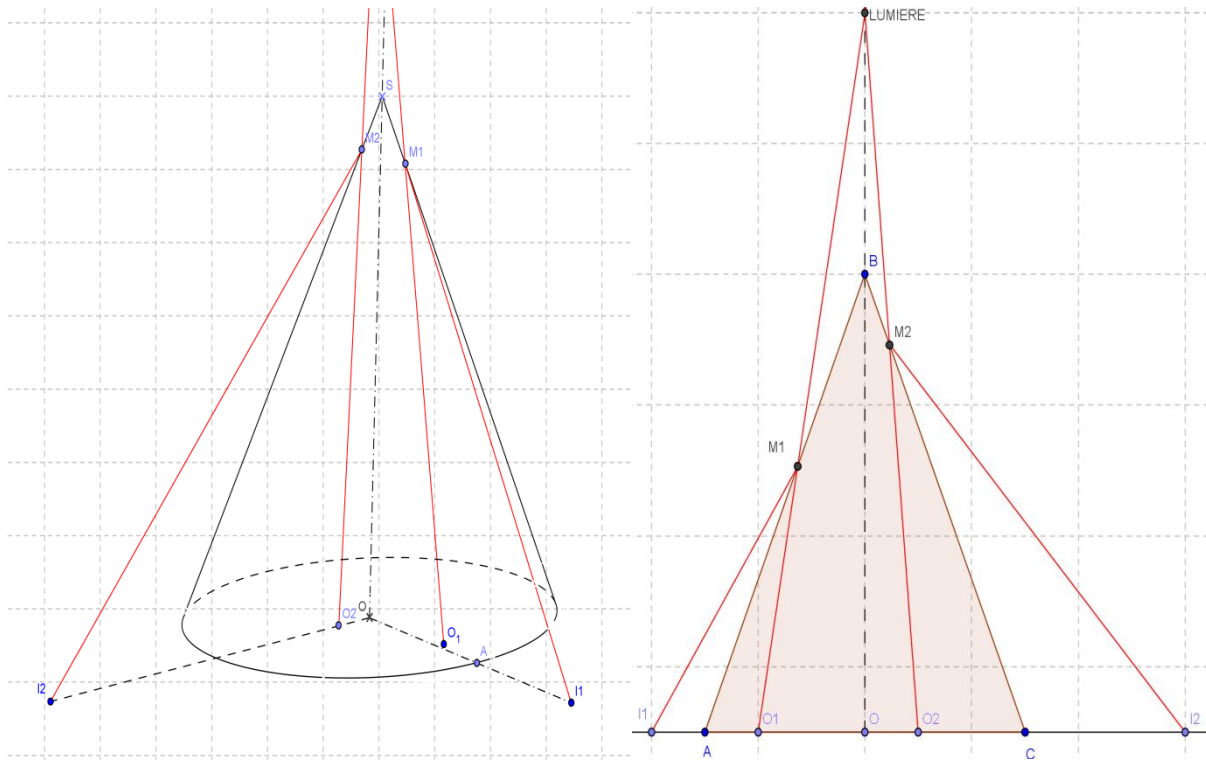
D'un point de vue plus contemporain, l'anamorphose est utilisée notamment dans le domaine militaire (l'exemple du périscope est le plus célèbre), dans le domaine de la cartographie, ou encore dans le domaine cinématographique... Elle possède ainsi encore de nombreuses utilités de nos jours, malgré sa simplicité de création.



3.1.2. Fonctionnement scientifique

3.1.2.1. Fonctionnement de base

Pour obtenir une anamorphose conique, il suffit de poser un cône de révolution réfléchissant sur un plan. Ce plan est également le lieu où est tracée la figure déformée. Elle sera ainsi reformée sur le cône. L'œil doit être placé au-dessus du sommet du cône c'est-à-dire sur l'axe de révolution de ce dernier.

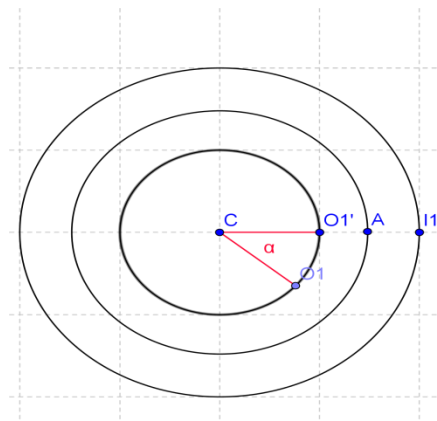


Le problème consiste à déterminer la figure réformée F' telle que F soit connue initialement.

Traisons le problème de manière graphique.

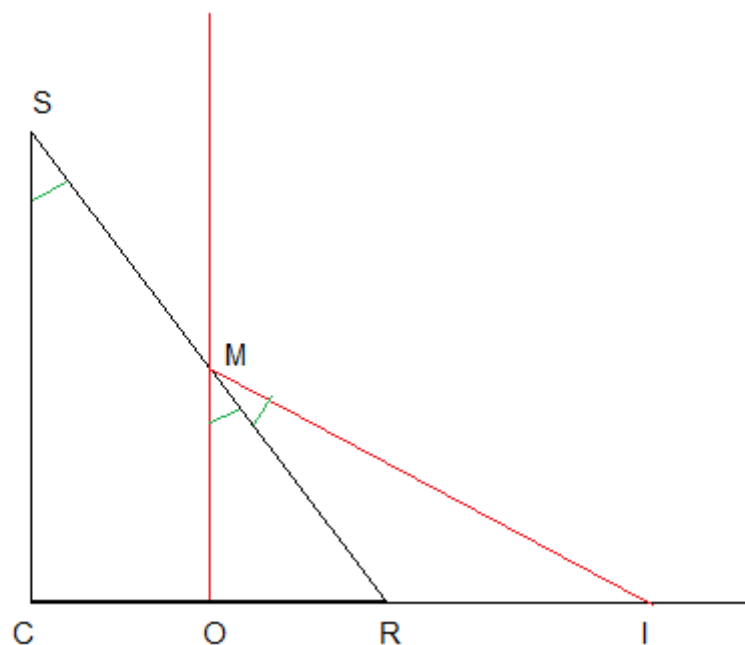
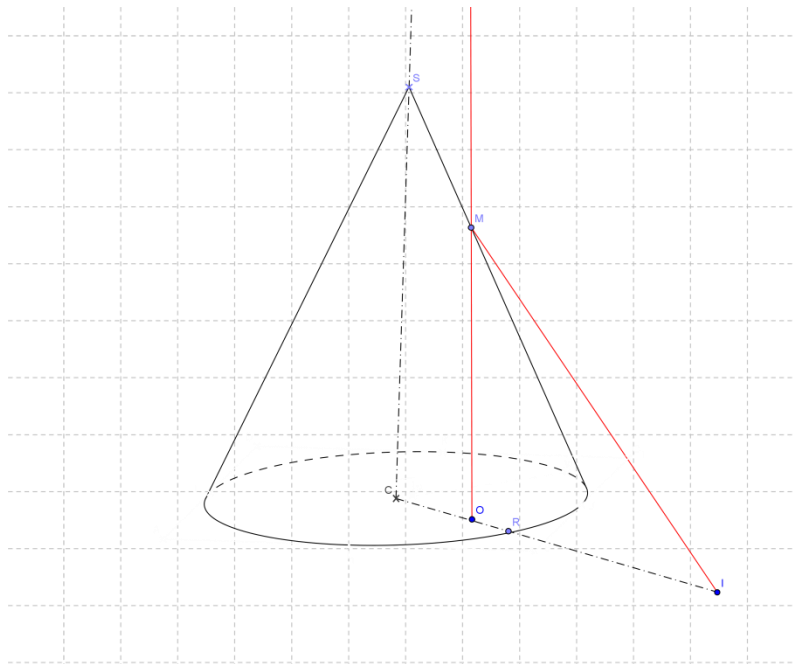
Soit O_1 un point du plan horizontal. Cherchons ce point tel qu'un rayon lumineux issu de O_1 soit réfléchi suivant la droite O_1M_1 . Les lois de la réflexion fournissent facilement une solution pour les points O_1 situés dans le plan vertical contenant l'axe de révolution du cône.

Pour un point O_1 non situés dans ce plan vertical, il suffit de se ramener dans le cas général grâce à une rotation d'axe l'axe de révolution du cône et d'angle α où α représente l'angle entre le plan vertical et O_1 . Grâce à cette rotation, on obtient l'ensemble des points O_1 recherchés.



3.1.2.2. Cas particulier : l'œil est placé à l'infini

Considérons les figures suivantes



Les rayons lumineux arrivent parallèles à l'axe de rotation du cône. On obtient alors une relation entre l'angle du cône et la distance de déformation des différents points de la figure à déformer.

Montrons que le rapport entre IR et OR est constant.

Notons respectivement maintenant d' , d et h les distances IR, OR, IM. ω définit quant à lui l'angle CSP.

Comme le montre la figure ci-dessus, les angles OMR, RMI sont égaux et valent eux deux ω . De plus, les triangles OMR et OMI sont tous deux rectangles en O.



On a grâce à la trigonométrie, $\tan(2\omega) = \frac{d+d'}{h}$ et $\tan(\omega) = \frac{d}{h}$.

En faisant le rapport entre ces deux équations, on obtient $\frac{\tan(2\omega)}{\tan(\omega)} = \frac{d+d'}{d} = 1 + \frac{d'}{d}$.

Utilisons maintenant les formules de trigonométrie pour simplifier cette expression.

On sait que $\tan(2\omega) = \frac{2 \tan(\omega)}{1 - \tan^2(\omega)}$

D'où, $\frac{\tan(2\omega)}{\tan(\omega)} = \frac{2}{1 - \tan^2(\omega)}$

On cherche à exprimer le rapport $\frac{d'}{d}$. Ainsi, en utilisant le rapport précédent, on obtient l'expression suivante :

$$\frac{d'}{d} = \frac{2}{1 - \tan^2(\omega)} - 1 = \frac{1 + \tan^2(\omega)}{1 - \tan^2(\omega)}$$

Connaissant tous nos formules de trigonométrie sur le bout des doigts, nous savons que

$$\frac{1 + \tan^2(\omega)}{1 - \tan^2(\omega)} = \frac{1}{\cos(2\omega)}$$

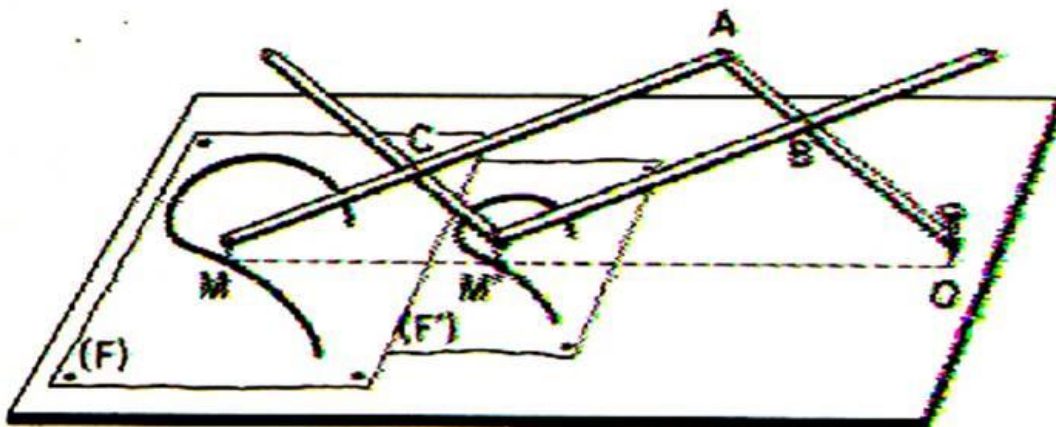
On obtient ainsi le rapport $\frac{d'}{d} = \frac{1}{\cos(2\omega)}$

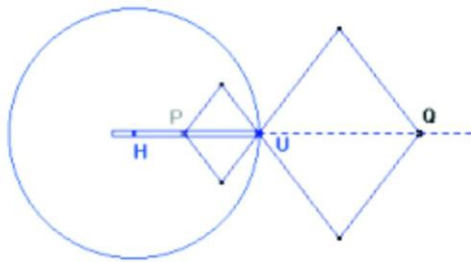
Ce rapport est donc bien constant, et la construction de l'image déformée s'effectue grâce à ce dernier.

3.1.3. Construction d'un pantographe

3.1.3.1. Historique

A l'origine le pantographe est un instrument de dessin, composé de tiges articulées qui permet de faire des agrandissements ou des réductions grâce aux propriétés de l'homothétie pour conserver les proportions entre le dessin original et la copie.





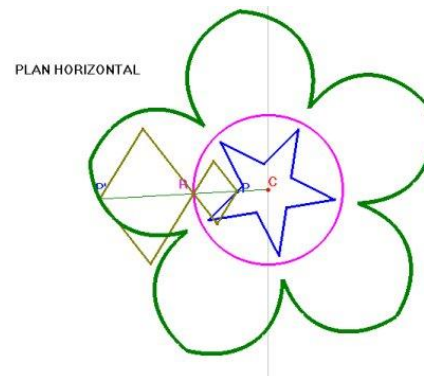
Le premier pantographe a été construit en 1603 par Christoph Scheiner pour recréer des diagrammes. Comme on peut le voir sur l'image ci-dessus, un premier bras (OA) est fixe par rapport au support, le bras central (M'B) est prolongé par un petit pointeur, et le dernier (MA) est muni d'un crayon. En déplaçant le pointeur sur l'image originale, une copie de cette dernière est réalisée par le crayon sur une autre feuille de papier. La dimension de l'image produite peut être modifiée en changeant la dimension du parallélogramme BACM'.

Encore utilisé il y a quelques dizaines d'années (par les architectes par exemple), cet outil est devenu obsolète à cause de l'apparition des outils informatiques.

Il existe aussi un autre type de pantographe qui va nous intéresser plus particulièrement. C'est le pantographe conçu par Michel PARRE en 1973 permettant de tracer mécaniquement les anamorphoses coniques observées de l'infini. L'ensemble consiste en un pantographe tournant autour du centre de la base du cône et qui centré sur le cercle, maintient un rapport constant entre la distance au cercle du point de l'image déformée et la distance au cercle du point de l'image originale.

Ce pantographe permet de réaliser de façon simple des anamorphoses pour miroirs coniques. Il consiste en une fente rectiligne HU qui est liée au pivot H de sorte que $HU = a$ (rayon de base du miroir conique). L'instrument est composé de deux losanges articulés (six tiges sont composées comme dans la figure), qui ont le sommet U en commun et deux extrémités P et Q opposées par rapport au point U. Le sommet P est contraint à coulisser dans la fente HU, qui pivote autour de H. Les sommets P et Q sont ainsi alignés avec U.

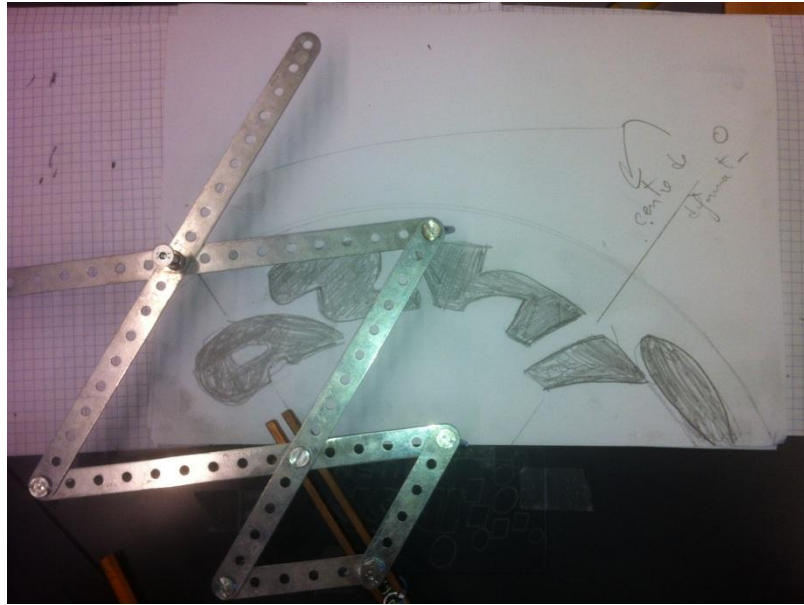
Si le rapport entre les côtés des deux losanges est choisi de façon appropriée, lorsque le point P décrit la figure placée à l'intérieur du cercle de base du cône, le point Q trace automatiquement l'image transformée.



3.1.3.2. Réalisation d'un pantographe en mécano

Dans le cadre de notre projet, nous avons voulu réaliser un pantographe afin de vérifier les anamorphoses obtenues par ordinateur à l'aide de nos algorithmes de déformation. Pour fabriquer cet outil, nous avons utilisé des mécanos fournis par notre professeur. En effet, l'utilisation de mécanos semblait suffisante pour la réalisation d'un pantographe; celui-ci étant en grande majorité constitué de simples barres et de vis pour articuler ces dernières. Mais pendant la fabrication, nous avons eu quelques problèmes. Un des problèmes rencontrés concernait la liaison glissière nécessaire au bon fonctionnement du pantographe, et qui ne pouvait être réalisé en mécano. Notre professeur nous a donc fabriqué une liaison glissière. Ensuite, une fois le pantographe fini, nous avons remarqué que ce dernier n'était pas très stable. En conséquence nous avons utilisé des vis de plus gros diamètre pour les articulations entre les barres, et nous avons fixé le tout sur une grande plaque support. Enfin, nous avons gravé notre image à déformer dans une fine plaque de PVC afin de pallier le manque de stabilité d'une feuille en papier.



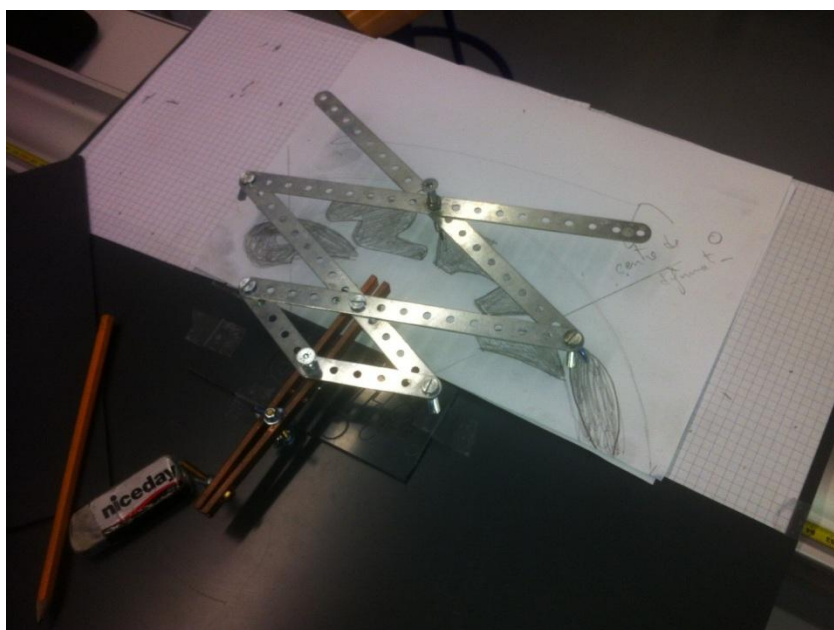


3.1.3.3. Application

Pour tester notre pantographe et vérifier nos anamorphoses, nous avons décidé de déformer le logo INSA qui est une forme assez simple; car pour le choix de l'image originale il y avait certaines restrictions dû aux limites de l'application. En effet notre pantographe ayant un rayon de déformation de 7.5 cm, l'image devait pouvoir tenir dans un cercle de 14 cm de diamètre. De plus nous n'avons pas placé l'image au centre du cercle pour avoir le meilleur rendu possible. Lors de l'utilisation du pantographe, étant donné que ce dernier est assez archaïque, nous avons dû faire plusieurs essais afin que le contour des lettres soit discernable. Enfin, pour obtenir les caractéristiques du cône réfléchissant qu'il fallait utiliser pour observer notre anamorphose, nous avons utilisé la formule décrite précédemment :

$$\frac{d'}{d} = \frac{1}{\cos(2\omega)}$$

Nous avons donc trouvé qu'il fallait utiliser un cône de rayon 7.5 cm et d'angle 19°.

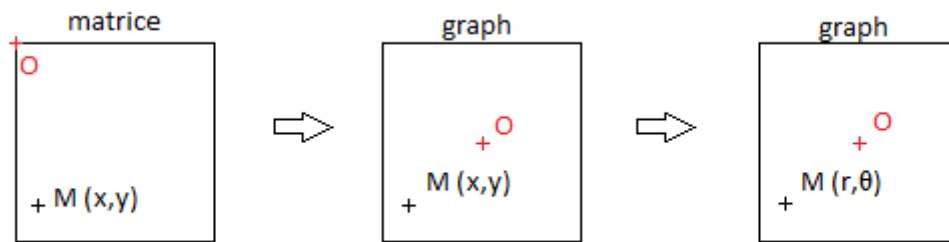


3.1.4. Programme permettant de créer ces anamorphoses

3.1.4.1. Principe

On transforme tout d'abord l'image en une matrice, où à chaque élément de coordonnées i,j de la matrice est associé la couleur du pixel situé aux mêmes coordonnées sur l'image.

Ensuite, on cherche à appliquer la formule que nous avons calculée pour trouver la position d'un point de l'image par une anamorphose conique. Une solution serait donc de se placer au centre de l'image pour pouvoir appliquer, en coordonnées polaires (r,θ) , cette modification de r , en faisant varier θ .



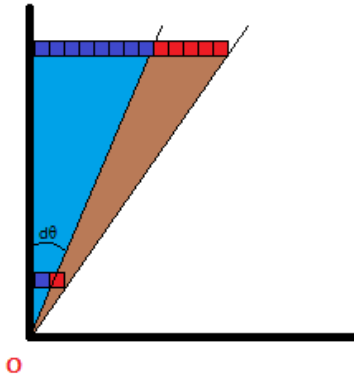
Après cela, on place notre dernier graphique au centre d'un autre graph plus grand, où on y ajoute les projections des points de l'image. On obtient alors quelque chose comme ceci :



Le résultat est là, mais il est clair que nous pourrions le perfectionner. On remarque qu'on a des problèmes de remplissage, on va donc prendre plus de valeurs de θ , ainsi on obtient :



Le résultat est déjà bien meilleur ! Cependant, on remarque que les bords de l'image ne sont toujours pas correctement remplis. On décide donc d'insérer en plus les variables $d\theta$ et $\Delta d\theta$, où $d\theta$ est la différence d'angle engendré par la largeur du pixel et $\Delta d\theta$ la sous-partie de $d\theta$ déterminée par le nombre de mesures prises entre θ et $\theta + d\theta$, l'intérêt de ces variables est donné ci-dessous :



Dans cet exemple, on obtiendrait un remplissage complet de la zone bleue avec un $\Delta d\theta$ au moins 8 fois plus petit que $d\theta$.

Dans le but de s'assurer un bon résultat, on utilise en plus de ces deux variables la fonction « strel », qui diffuse la couleur d'un pixel sur les pixels voisins à ce dernier. Et à la fin, on obtient :



Vous trouverez le code de ce programme expliqué et commenté à la fin de ce rapport.

3.1.5. *L'utilisation du logiciel Pov-Ray*

Povray (The Persistence of Vision Raytracer) est un logiciel de 2009 qui permet de créer des images de synthèse en des visualisations d'image 3 dimensions en 2 dimensions à partir d'un langage de programmation. C'est un logiciel libre, donc gratuit disponible pour de nombreuses plateformes et dont on peut consulter et modifier son code source.

Nous l'utilisons pour créer informatiquement une anamorphose à partir d'une image transformée grâce au code Matlab. Notre objectif est alors de retrouver l'image initiale dans le miroir conique généré par Povray. Nous devons donc faire attention à la position de l'image et du miroir, au rayon et à l'angle pour le cône.



Au niveau du code, nous avons commencé par tâtonner afin de prendre connaissance du logiciel. Notre but étant l'anamorphose conique.

- Première étape : positionner la caméra et la source de lumière, sans quoi l'image obtenue serait noire.
- Deuxième étape : définir un plan en $z=0$ (z correspond à la hauteur) au départ d'une seule couleur,
- Troisième étape : Insérer une image.
- Quatrième étape : Créer le cône (d'abord simple puis réfléchissant), défini par sa hauteur et son rayon.

Une fois arrivé à des résultats convenables, nous nous sommes demandé comment généraliser notre code. En effet, nous ne travaillions que sur des cas particuliers d'image obtenue par le logiciel AnamorphMe (qui transforme une image pour obtenir une anamorphose).

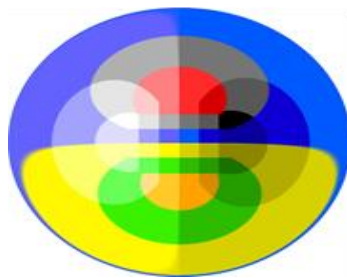


Figure 2: Image obtenue grâce à AnamorphMe

Pour que le code soit valable quel que soit le cône, il fallait mettre en entrée les paramètres du cône (le rayon en pixel et l'angle en degré) correspondant aux données utilisées par le programme en Matlab. Les échelles de taille n'étant pas les mêmes, nous avons également dû trouver le rapport entre le rayon « povray » et le rayon en pixel. Pour déterminer la hauteur du cône, il nous a suffi de prendre le rayon divisé par la tangente de l'angle en radian.



Figure 3 : Anamorphose conique du logo INSA grâce à Pov-Ray

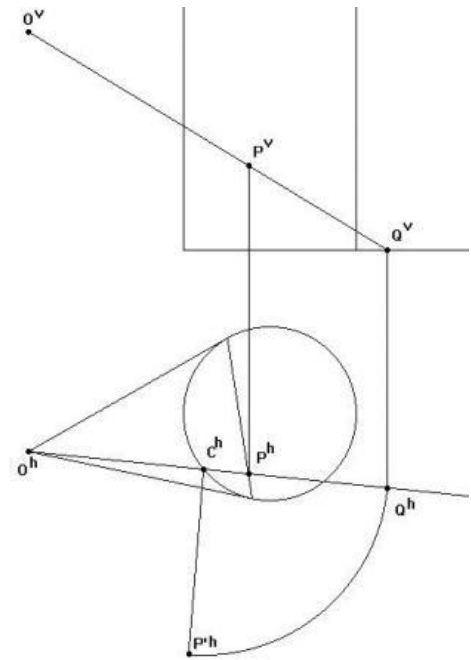


3.2. L'anamorphose cylindrique

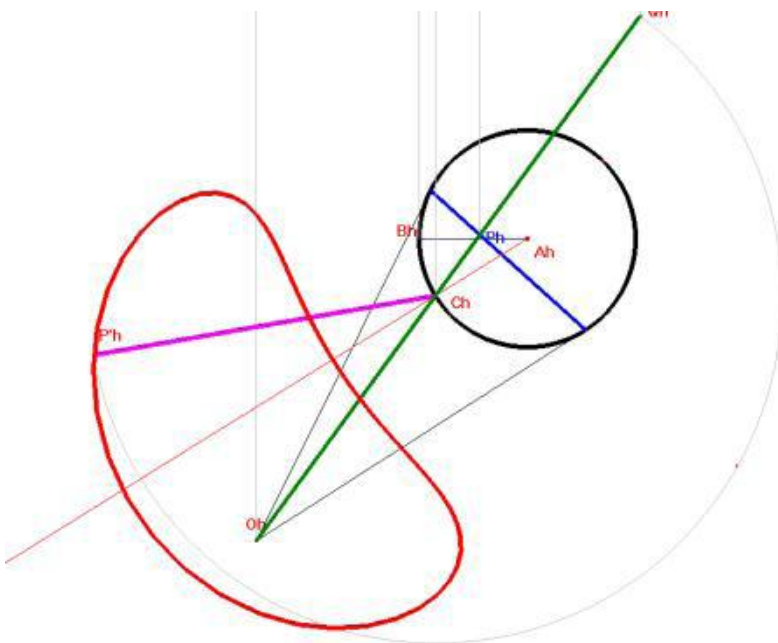
3.2.1. Fonctionnement scientifique

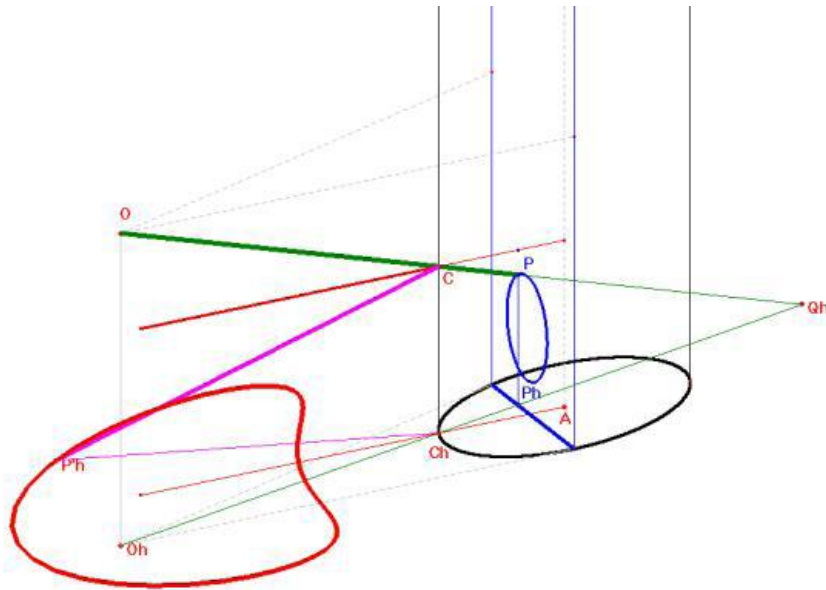
Afin d'obtenir une anamorphose cylindrique, il faut se munir d'un cylindre de révolution avec une surface réfléchissante et le poser sur un support plan, afin qu'il soit perpendiculaire à ce dernier. Si une image déformée est dessinée sur le plan, l'observateur verra l'image non déformée en regardant d'un point de vue donné le miroir cylindrique.

Essayons de déterminer un point P' de l'image déformée connaissant un point P de l'image non déformée. Supposons qu'un cylindre réfléchissant soit posé sur un plan horizontal de projection et que l'œil soit placé en O . Les images situées dans le plan horizontal semblent se réfléchir dans le plan de contact déterminé par les plans tangents issus de O . Soit un point P dans ce plan de contact. Nous devons maintenant trouver un point P' du plan horizontal tel qu'un rayon lumineux issu de P' soit réfléchi suivant la droite PO .



Soit Q le point d'intersection de OP dans le plan horizontal de projection. La surface cylindrique est traversée par la droite OP en deux points. Soit C celui situé entre O et P . Les distances $C^h P'^h$ et $C^h Q^h$ devant être égales, le point P' est complètement déterminé par le fait que $O^h C^h$ et $P'^h C^h$ forment des angles égaux avec la tangente en C^h au cercle de base. Si une image déformée est dessinée sur le plan, l'observateur verra l'image non déformée en regardant d'un point de vue donné le miroir cylindrique.





3.2.2. Programme permettant de créer ces anamorphoses

Lors de la programmation de l'anamorphose cylindrique, nous nous sommes tenus à la même logique que pour la conique, aussi, cette fois-ci, nous avons anticipé les problèmes rencontrés auparavant en intégrant dès la première version les variables $d\theta$ et $\Delta d\theta$ ainsi que la fonction « strel », les seules différences notables entre les deux codes sont, d'abord, la formule qui sert à calculer la nouvelle distance d'un pixel par rapport à l'origine après la transformation, et ensuite, l'organisation du code, car nous sommes allés plus vite la deuxième fois.

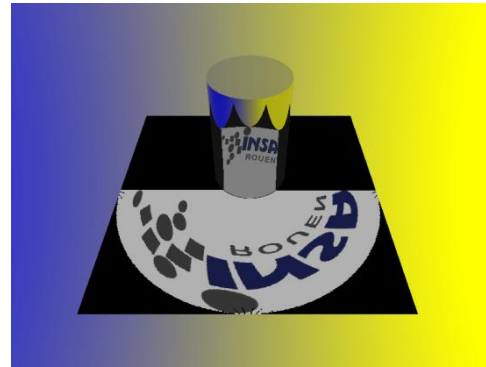
Et voici ce que nous obtenons :



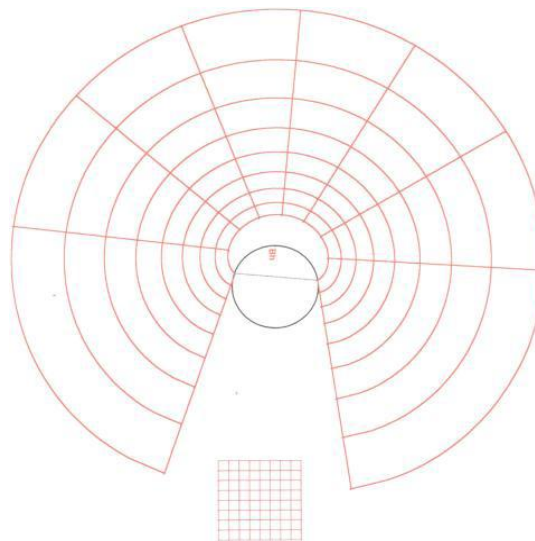
3.2.3. Utilisation de Pov-Ray

En utilisant le logiciel PovRay pour reconstituer une image déformée en cylindrique, nous avons remarqué une particularité : dans l'image finale donnée par le miroir, les lignes qui devraient être parfaitement horizontales apparaissent légèrement courbées. Pour rectifier cela, nous avons un petit peu excentré la position du cylindre par rapport au centre du cercle que forme notre image déformée. Alors, le défaut de parallélisme semble se résoudre. Il nous fallait alors expliquer les raisons de ce phénomène.

Sur l'image ci-contre, on voit que le logo INSA n'apparaît pas parfaitement horizontal dans le cylindre réfléchissant.

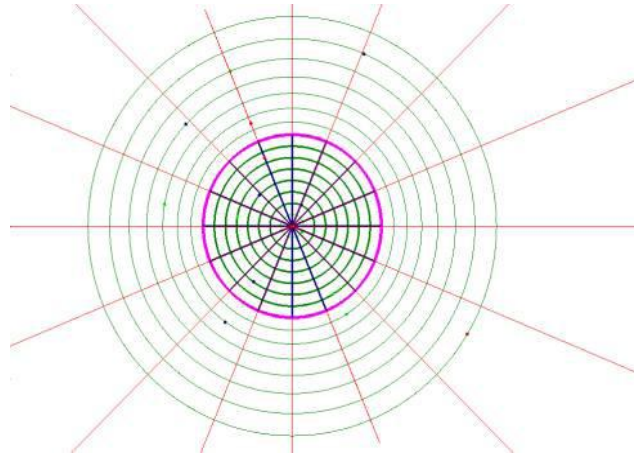


En réalité, notre méthode pour déformer les images en cylindrique avec le programme informatique que nous avons conçu souffre d'une erreur d'approximation. En effet, c'est directement la déformation (r, θ) qui doit être revue. En pratique, les images déformées en coniques semblent prendre la forme d'un cercle qui s'enroule autour du cylindre, et nous avons construit notre programme dans ce sens. Cependant, en étudiant le phénomène, nous nous sommes rendus compte qu'il ne s'agit que d'une approximation, qui bien que facilitant grandement les calculs et la création du programme informatique, n'est pas le reflet exact de la réalité.



Grille de construction donnant des droites parallèles en cylindrique. On peut remarquer que les cercles apparents ne sont pas parfaits, et que le miroir n'est pas centré.



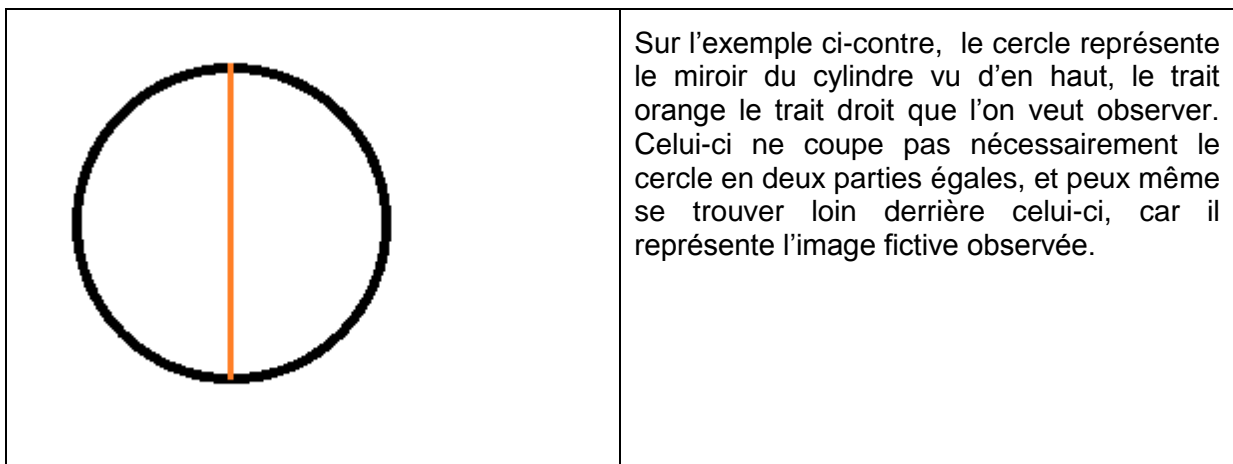


Nous n'avons pas eu de problèmes similaires pour les anamorphoses coniques, pour lesquelles la grille des droites parallèles forme des cercles parfaits, et nécessite un cône placé au centre.

Pour se rendre compte de cela, il faut étudier la trajectoire des rayons lumineux. On se rend compte alors que pour voir un trait droit, il faut que l'image initiale ne soit pas circulaire, mais elliptique. Notre erreur d'appréciation vient du fait que l'ellipse en question a une forme quasi-circulaire ! Prenons l'exemple suivant :

Nous voulons que l'image restituée soit vue parfaitement droite par l'observateur.

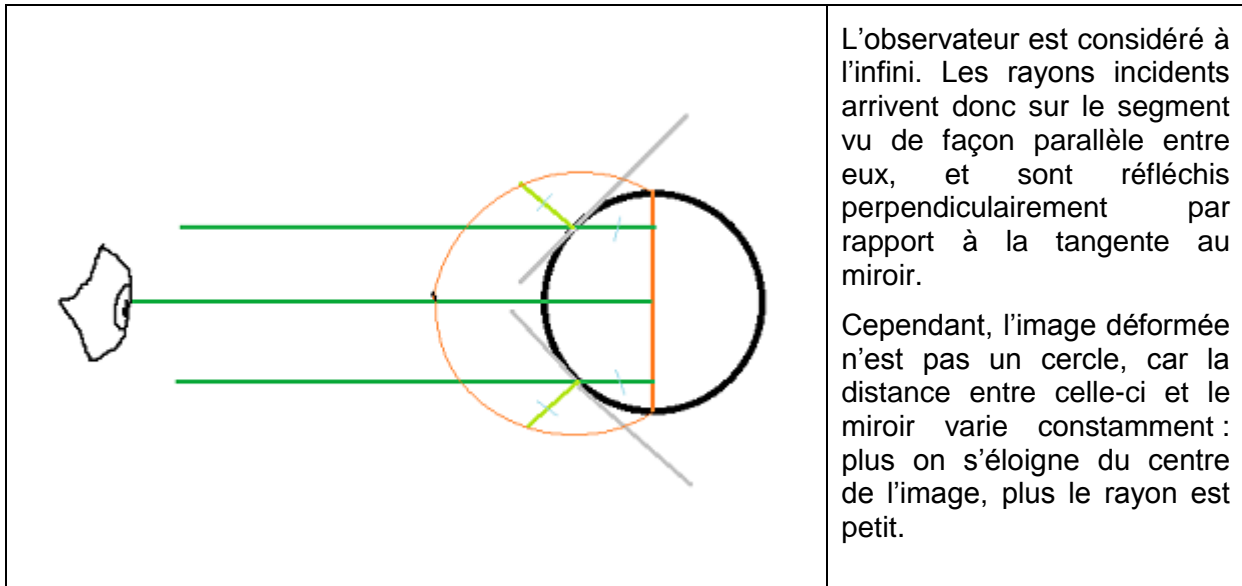
Sur l'exemple ci-contre, le cercle représente le miroir du cylindre vu d'en haut, le trait orange le trait droit que l'on veut observer. Celui-ci ne coupe pas nécessairement le cercle en deux parties égales, et peut même se trouver loin derrière celui-ci, car il représente l'image fictive observée.



L'observateur est considéré à l'infini. Les rayons incidents arrivent donc sur le segment vu de façon parallèle entre eux, et sont réfléchis perpendiculairement par rapport à la tangente au miroir.

Cependant, l'image déformée n'est pas un cercle, car la distance entre celle-ci et le miroir varie constamment : plus on s'éloigne du centre de l'image, plus le rayon est petit.





L'image ci-dessus nous permet de voir que la distance entre l'image déformée et le miroir cylindrique n'est pas constante : elle vaut $r \cdot \cos(\alpha)$, avec r le rayon du miroir cylindrique, et α l'angle entre le rayon incident et le rayon réfléchi.

L'image réfléchi et donc en réalité de forme elliptique, dont les deux foyers sont très proches l'un de l'autre, et pourtant pas confondus (d'où la forme proche d'un cercle).

Pour obtenir un résultat optimal, il nous faudrait un programme qui restitue la forme elliptique de l'image déformée. Cependant, pour des raisons de difficultés, nous avons décidé de garder notre logiciel, qui nous déforme l'image initiale de façon circulaire.

Pour compenser ce défaut de construction, nous avons pensés à reculer légèrement le miroir cylindrique par rapport à l'image. Ainsi, le décalage entre le centre du miroir et le centre de l'image donne une illusion approximative d'une ellipse, malgré le fait que les deux éléments soient circulaires. L'image observée nous apparaît alors satisfaisante, même si des imperfections peuvent être visibles sur les bords de l'image.



4. CONCLUSIONS ET PERSPECTIVES

4.1. Conclusions sur le travail réalisé

Ce projet nous a permis de découvrir le fonctionnement des anamorphoses cylindriques et conique et d'en recréer d'un point de vue mécanique, en utilisant des logiciels déjà conçus (pov-ray et anamorph-me) et en créant nos propres algorithmes d'anamorphose (avec matlab)

4.2. Conclusions sur l'apport personnel de cet E.C. projet

Tout d'abord, cette U.V. nous a permis de nous apprendre et habituer à travailler en équipe sur un projet donné comme on peut trouver dans le monde professionnel. La réalisation d'un tel projet demande de l'intérêt et de l'investissement personnel s'il veut être mené à bien.

D'autre part, ce projet a permis d'approfondir nos connaissances dans plusieurs domaines scientifiques : de l'optique géométrique, d'algorithmique et de gestion de logiciels. Apprendre à programmer en Matlab a également permis d'avoir certaines bases pour plusieurs étudiants du groupe souhaitant effectuer les départements Génie Mathématique et Architecture des Systèmes d'Informatiques.

Même si le travail a été laborieux et long, il ressort de ce projet un bilan positif tant au niveau du résultat que de l'apport personnel.

4.3. Perspectives pour la poursuite de ce projet

Par manque de temps, le programme créé n'est pas parfait. En effet, on modifie l'image en utilisant seulement deux paramètres r et θ . Et, on a considéré r comme constant pour une même ligne alors que l'on devrait insérer une fonction $dr = f(\theta)$. En effectuant ceci, l'image restituée serait exacte.



5. BIBLIOGRAPHIE

Sites collaboratifs :

- [http://fr.wikipedia.org/wiki/Pantographe_\(dessin\)](http://fr.wikipedia.org/wiki/Pantographe_(dessin))
- <http://fr.wikipedia.org/wiki/Anamorphose>

Site d'aide scientifique :

- <http://users.skynet.be/mathema/pantog.htm>
- <http://archiviomacmat.unimore.it/PAWeb/Sito/Francese/263f.htm>
- http://www.ac-noumea.nc/maths/IMG/pdf/Expe_rience_en_Sixie_me_et_Cinquie_me.pdf
- <http://membres.multimania.fr/anamorphs/cylindrique.htm>
- <http://people.via.ecp.fr/~jm/TTLanamorphose/geometrie.htm>

6. ANNEXES

6.1. Enoncé des logiciels utilisés

- Maple 15 pour tracer des ellipses,
- Matlab pour programmer les anamorphoses,
- Pov Ray pour reformer une image déformée à l'aide de jeux de miroir,
- Anamorphme pour comprendre le fonctionnement des anamorphoses.
- GeoGebra pour tracer les figures du rapport

6.2. Listings des programmes réalisés

6.2.1. *Calcul d'une anamorphose conique*

1. Entrée des différents paramètres nécessaires à l'anamorphose

rayon=190

angle2=30

angle2rad=angle2*pi/180

hauteur=rayon/tan(angle2rad)

angle=2*angle2rad

pasmax=20

remplissage de l'image

rayon du cône en pixel

demi angle au sommet du cône en degrés

calcul du demi angle en radian

calcul de la hauteur du cône

calcul de l'angle au sommet du cone

nombre maximum de compartimentation pour le

2. Lecture et dimensionnement de l'image




```
A = imread('C:\Users\Matthieu\Desktop\Cours\STPI2 - 4eme semestre (GM-ASI)\P6 (Anamorphoses)\insa2 - Copie.jpg'); lecture de l'image depuis l'ordinateur
imagesc(A) affichage de l'image dans Matlab
dimx=size(A,1) la constante dimx prend la taille de la largeur de l'image (abscisse)
dimy=size(A,2) la constante dimy prend la taille de la longueur de l'image (ordonnée)
```

3. Rayon du disque réfléchi

```
r2=hauteur*tan(angle) longueur de la génératrice du cône projetée sur le plan. C'est donc la longueur du point le plus éloigné du centre après l'anamorphose
dim2=round(r2) dim2 prend la valeur arrondie de la génératrice
```

4. Initialisation de la matrice de destination

```
dim3=dim2*2 dim3 prend deux fois la longueur de la génératrice
AA=zeros([dim3 dim3]); Initialisation de la matrice AA de taille dim3 en abscisse et dim3 en ordonnée remplie de zéros
```

5. Coefficient d'agrandissement du rayon

```
k=(r2-rayon)/rayon; quel que soit le point de la matrice, après transformation, le nouveau rayon sera proportionnel à l'ancien.
```

6. Balayage de la surface et calcul des coordonnées des points après l'anamorphose

```
for x=1:dimx boucle dans laquelle x prend toutes les valeurs de 1 a dimx (x prend toutes les valeurs en abscisse de l'image)
for y=1:dimy boucle dans laquelle y prend toutes les valeurs de 1 a dimy (y prend toutes les valeurs en ordonnée de l'image)
xcentered=x-dimx/2; recentrage des coordonnées en x de l'image dans la nouvelle matrice
ycentered=y-dimy/2; recentrage des coordonnées en y de l'image dans la nouvelle matrice
[theta,r]= cart2pol(xcentered,ycentered); transformation des coordonnées en x et y recentré en coordonnées polaires (theta et r)
if r<=rayon condition sur r qui doit être inférieur au rayon
rr=rayon+k*(rayon-r); calcul des nouvelles coordonnées du rayon après l'anamorphose.
On utilise le coefficient de grandissement calculé plus tot
```

7. Remplissage des trous

```
dtheta=atan(1/r); calcul de l'angle pour un pixel
deltadtheta=dtheta/pasmax; calcul de l'angle de chaque compartimentation
for pas=1: (pasmax+1) boucle dans laquelle le pas va de 1 au nombre de compartimentation plus 1
tetaremplissage=theta+deltadtheta*(pas-1); calcul des différents angles des pixels de remplissage pour un même rayon
[xremp,yremp]=pol2cart(tetaremplissage,rr); passage des coordonnées polaire (en tetaremplissage : l'angle et rr : le nouveau rayon après anamorphose) en coordonnées cartésiennes
xrempdecentre=round(xremp+dim3/2); décentrage des nouvelles coordonnées cartésiennes en x
yrempdecentre=round(yremp+dim3/2); décentrage des nouvelles coordonnées cartésiennes en y
for kkk=1: 3 boucle ou kkk prend les valeurs de 1 à 3 (ce qui correspond aux couleurs rouge vert bleu)
AA(xrempdecentre,yrempdecentre,kkk)=A(x,y,kkk); Les valeurs (et donc les couleurs) de la nouvelles matrices des coordonnées en x et y décentrés prennent les même valeurs que celles de la matrice image initiale.
end
end
```

8. Image du centre

```
[newx,newy]=pol2cart(theta,r); passage des coordonnées polaires de l'image initiale en coordonnées cartésiennes
```



```

new2x=round(newx+dim3/2); décentrage des nouvelles coordonnées en x de la matrice
initiale
new2y=round(newy+dim3/2); décentrage des nouvelles coordonnées en y de la matrice
initiale
for kkk=1:3 Boucle des valeurs rouge vert bleu
AA(new2x,new2y,kkk)=A(x,y,kkk); Les valeurs de la matrices des nouvelles coordonnées
en x et y prennent les même valeurs que celles de la matrice image initiale. Cela permet de voir
l'image initial au centre de l'image après anamorphose.
end
end
end
end

```

9. Amélioration de l'image (remplissage des trous restants)

```

AFinal2 = uint8(round(AA- 1)); converti la matrice en format image (8 bits)
imagesc(AFinal2); affichage de l'image
SE = strel('rectangle',[3 3]); créé un élément structurant de forme rectangulaire de
dimension trois pixels sur trois pixels
BW2 = imdilate(AFinal2,SE); l'élément structurant passe sur toute les frontières de
couleurs de l'image et rebouche les trous en donnant la même couleur aux huit pixels autour de
celui sur lequel il se trouve
BW2 = imerode(BW2,SE); Corrige les pixels qui dépassent aux frontières par un nouveau
passage de l'élément structurant sur l'image.

```

10. Traçage d'un cercle sur l'image pour bien positionner le cylindre miroir

```

for x=1:dim3 boucle dans laquelle x prend toutes les coordonnées en abscisse de l'image
for y=1:dim3 boucle dans laquelle y prend toutes les coordonnées en ordonnée
de l'image
xcentered=x-dim3/2; recentrage des coordonnées en x de l'image dans la matrice
destination
ycentered=y-dim3/2; recentrage des coordonnées en y de l'image dans la matrice
destination
[theta,r]= cart2pol(xcentered,ycentered); transformation des coordonnées en x et y
recentré en coordonnées polaires (theta et r)
if r > rayon-1 && r < rayon+ 1 condition sur r pour que celui ci soit égal au rayon. Les
points qui respectent cette condition forment un cercle ayant pour rayon le rayon du cône
[newx,newy]=pol2cart(theta,r); passage des coordonnées polaires du cercle en
coordonnées cartésiennes
new2x=round(newx+dim3/2); décentrage des nouvelles coordonnées cartésiennes en x du
cercle
new2y=round(newy+dim3/2); décentrage des nouvelles coordonnées cartésiennes en y du
cercle
for kkk=1:3 boucle dans laquelle kkk va de 1 a 3 (valeurs rouge vert bleu)
BW2(new2x,new2y,kkk)=50; Les pixels du cercle quelle que soit leur couleur initiale prennent
la couleur grise
end
end
end
end

```

```

imshow(BW2) affichage de l'image

```

```

imwrite(AFinal,'C:\Users\varea.YELLOWCARD\Desktop\povr\insalogomatlab.png')
nommage et enregistrement de l'image sur l'ordinateur

```



6.2.2. Calcul d'une anamorphose cylindrique

Entrée des différents paramètres nécessaires à l'anamorphose

rayon=500 **rayon de déformation de l'image en pixels**
 angle=360 **angle de déformation de l'image en degrés**
 angle=angle/180*pi **angle de déformation de l'image en radians**

A = imread('C:\Users\seb\Downloads\logoinsaderouen.jpg'); **lecture de l'image depuis l'ordinateur**

imagesc(A) **affichage de l'image dans matlab**
 dimx=size(A,1) **dimx prend la taille de la largeur de l'image (abscisse)**
 dimy=size(A,2) **dimy prend la taille de la longueur de l'image (ordonnée)**

Matrice de destination

dim3=(dimx+rayon) **dim3 prend la taille de dimx plus le rayon de déformation**

Balayage de la surface de destination

for x=1:dimx **boucle allant de 1 a dimx (avec un pas de 1)**
 for y=dimy:-1:1 **boucle décroissante allant de dimy a 1(avec un pas de -1)**
 x **affichage de la variable x sur matlab pour savoir ou en est la transformation**
 r=rayon+dimx+dimx/(dimx-1)*(1-x); **calcul de la distance entre chaque pixel de l'image et du centre du cylindre**
 grandissement= r*angle/dimy; **calcul du coefficient de grandissement pour chaque pixel (ce qui donnera la nouvelle distance entre chaque pixel et le centre du cylindre)**
 dTheta= grandissement/r *(y-1); **calcul de l'angle des nouvelles coordonnées après transformation pour chaque pixel**

[xnew,ynew]=pol2cart(dTheta,r); **passage des coordonnées polaires de l'image transformée en coordonnées cartésiennes**

xnew=round(xnew+dim3); **décentrage des nouvelles coordonnées cartésiennes en x**
 ynew=round(ynew+dim3); **décentrage des nouvelles coordonnées cartésiennes en y**

if xnew ~= 0 && ynew ~= 0 **condition sur xnew et ynew qui doivent être différents de zéro**

 for kkk=1:3 **boucle des valeurs rouge vert bleu**
 AA(xnew,ynew,kkk)=A(x,y,kkk); **les valeurs de la nouvelles matrices des coordonnées en x et y décentrés prennent les même valeurs que celles de la matrice image initiale.**

 end

end

end

end

SE = strel('rectangle',[4 4]); **créé un élément structurant de forme rectangulaire de dimension quatre pixels sur quatre pixels**

AAA = imdilate(AA,SE); **l'élément structurant passe sur toutes les frontières de couleurs de l'image et rebouche les trous en donnant la même couleur a tous les pixels autour de celui sur lequel il se trouve**

AAAA= imerode(AAA,SE); **corrige les pixels qui dépassent aux frontières par un nouveau passage de l'élément structurant sur l'image. Cela enlève les effets de flou.**

AFinal = uint8(round(AAAA- 1)); **converti la matrice en format image (8 bits)**

imagesc(AFinal) **affichage de l'image dans matlab**

imwrite(AFinal, 'C:\Users\seb\Downloads\logo-insaRouencylindrique2.jpg')
nommage et enregistrement de l'image sur l'ordinateur



6.2.3. Code Pov Ray en conique pour reformer l'image

```

#include "colors.inc"          permet d'utiliser les couleurs (gray, white ...)
#include "textures.inc"       permet d'utiliser les textures (finish : affecte
                             l'apparence d'une surface)
#include "glass.inc"         permet d'utiliser la réflexion (reflection)
#declare RayonEnPixel = 300 ; #déclare : pour déclarer une variable
#declare AngleEnDegre = 30 ;
#declare AngleEnRadian = AngleEnDegre * pi / 180 ;
#declare HauteurEnPixel = RayonEnPixel / tan(AngleEnRadian) ;
#declare Hauteur = HauteurEnPixel * 0.000566667 ;    0.000566667 est le rapport entre
                                                    pixel et Povray
#declare Rayon = RayonEnPixel * 0.000566667 ;    (300 pixels correspond à 0.17 en
                                                    Povray)

camera {location <0,0,1>      position et direction de la caméra (l'œil)
        look_at <0,0,0>}
light_source {<0,0,200> color White} position et couleur de la lumière
plane {z,0
  pigment {                  couleur de la surface
    image_map {              pour insérer une image
      png "C:\Users\Jeanne_2\Documents\POV-Ray\v3.6\include\insanamorph.png"
      once                    l'image n'est pas répétée à l'infini
    }
  }
  translate <-0.5,-0.5,0>    on tourne l'image de manière à ce qu'elle se place au
                                                    niveau du cône
}
cone {<0,0,Hauteur>,0      la pointe du cône a les coordonnées (0,0,Hauteur) et
                            le rayon = 0
      <0,0,0>Rayon          la base du cône a les coordonnées (0,0,0) et le rayon
                            = Rayon

  pigment { color Gray }    modifie l'apparence du cône pour le rendre grisé afin de voir le miroir
  finish { ambient .1 diffuse .1
    reflection { 0.8 metallic } rend le cône réfléchissant avec un effet métallique
  }
}
fog {White distance 550}    rend l'image plus lumineuse

```

6.2.4. Code Pov Ray en cylindrique pour reformer l'image

```

#include "colors.inc"          permet d'utiliser les couleurs (gray, white ...)
#include "textures.inc"       permet d'utiliser les textures (finish : affecte
                             l'apparence d'une surface)
#include "glass.inc"         permet d'utiliser la réflexion (reflection)

camera {location <0,0,-10>    position et direction de la caméra (l'œil)
        look_at <0,0,10>
        rotate <-45,0,90>
}
light_source {<0,0,-200> color White} position et couleur de la lumière
sky_sphere {                  Cette fonction (trouvée dans l'aide) permet de
  pigment {                    faire un fond dégradé bicolore, plus facile à
    gradient y                  exploiter.
    color_map {

```



```

    [(1-cos(radians( 30)))/2 color Blue]
    [(1-cos(radians(120)))/2 color Yellow]
  }
  scale 2
  translate -1
}
}
plane {z,0
  pigment {
    image_map {
      png "C:\Users\Jeanne_2\Documents\POV-Ray\v3.6\include\insa2_cylindrique.png"
      once
    }
    aspect de la surface
    pour insérer une image
    l'image n'est pas répétée à l'infini
  }
  scale (7)
  translate <-3.5,-3.55,0>
  on tourne l'image de manière à ce qu'elle se place au
  niveau du cylindre
}
cylinder {<-0.5,0,0>
  <-0.5,0,-3>
  1
  le bas du cylindre a les coordonnées (-0.5,0,0)
  le haut du cylindre a les coordonnées (0,0,0)
  le rayon est de 1

  pigment { color Gray }
  finish { ambient .1 diffuse .1
  modifie l'apparence du cylindre pour le rendre grisé

  reflection { 0.8 metallic }
  rend le cylindre réfléchissant avec un effet
  métallique
}
}

```

